IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	James R. Wason	**Examiner:**	Joshua D. Campbell
Appeal:	2007-1186	**Decided:**	June 18, 2007
Serial No:	09/616,809	**Filing Date:**	July 14, 2000
Docket:	END920000080US1 (13679)	**Dated:**	August 16, 2007

For: "TEXT FILE INTERFACE SUPPORT IN AN OBJECT ORIENTED APPLICATION"

Confirmation No.: 6597

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR REHEARING UNDER 37 C.F.R. §41.52

Sir:

In a Decision On Appeal (Decision) dated June 18, 2007, the Board of Appeals affirmed the final rejection of Claims 1-3 and 5-17 under 35 U.S.C. 102 as being fully anticipated by U.S. Patent 6,317,871 (Andrews, et al.). Appellant respectfully requests a rehearing of the issue of whether the rejection of Claims 14-16 was proper. Claim 14 is representative of Claims 15 and 16.

CERTIFICATE OF ELECTRONIC FILING

I hereby certify that this correspondence is being deposited with the United States Patent & Trademark Office via Electronic Filing through the United States Patent and Trademark Office e-business website, on August 16, 2007.

Dated: August 16, 2007

John S. Sensny
John S. Sensny

Appellant respectfully submits that in its Decision, the Board of Appeals misapprehended or overlooked the following points:

I. The proper test to determine anticipation under 35 U.S.C. 102; and

II. Whether Andrews, et al. passes to the macro class a name for another template, which is subsequently invoked by the macro class, when the macro class is invoked.

Each of these points is discussed below.

I. The proper test for anticipation

In its Decision, the Board, in its basic identification of the issue, stated that the "issue turns on the interpretation of claim 1 and whether Andrews describes or suggests each and every limitation of the claims" (Decision, page 5, lines 2-4) (emphasis added).

Under the law, however, in order to sustain a rejection under 35 U.S.C. 102, it is not sufficient that the reference merely describes or suggests each and every claim limitation. Instead, strict identity is required.

The Court of Appeals for the Federal Circuit emphasizes that a strict identity test must be met in order for a reference to anticipate a claim under 35 U.S.C. 102. For instance, in Apple Computer, Inc. v. Articulate Systems, Inc., 57 USPQ2d 1057, 1061 (Fed. Cir 2000), the Court explained that: "Anticipation under 35 U.S.C. 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention." "Substantial identity" or "equivalency" is not sufficient. RCA Corp. V. Applied Digital Data Sys., Inc., 221 USPQ 385 (Fed. Cir. 1984).

Applicant respectfully submits that use by the Board of the improper standard for determining anticipation under 35 U.S.C. 102, which is the basis for the Examiner's rejection, is a fatal flaw to the Decision by the Board to sustain that rejection. It is therefore requested that the Board reconsider under the proper, strict identity test the Decision to sustain the rejection of Claim 14. It is further submitted that, when that proper test is applied, Claim 14 is not anticipated by Andrews, et al. for the reasons set forth in Appellant's Brief and Reply Brief.

II. Andrews, et al. does not show passing to the macro class a name for another template, which is subsequently invoked by the macro class, when the macro class is invoked.

With respect to Claim 14, the Board stated that:

"Andrews teaches a macro invocation fragment invoking another partition template to further process the source language code file. As part of this process, Andrews discloses passing a name for another template when the macro is invoked. Specifically, Figure 6 shows the name of the partition template (part'n) invoked by the macro invocation fragment." (Decision, page 12, lines 12-17).

The Board further stated that:

"Andrews inherently discloses passing a name for another template when the macro class is invoked. In order for the macro invocation fragment to invoke the partition template, a person of ordinary skill in the art would recognize that the

name of the partition template must necessarily be passed to the macro invocation fragment." (Decision, page 12, lines 19-24.

Applicant's Attorney has carefully studied Andrews, et al; and on the basis of that study, it is respectfully submitted that this reference does not disclose, expressly or inherently, passing to the macro the name of that other template when the macro class is invoked.

In order to best understand this difference between the present invention and Andrews, et al, it may be helpful to review briefly this invention and Andrews, et al.

This Invention

The present invention relates to a method and system for processing text files in computer application. In accordance with this invention, a plurality of templates are formed having literal fragments of the text file, and one or more macro classes are provided to map data from the text file to the computer application. Pointers to the macro classes are embedded in the templates. In operation, a template is used as an overlay to parse the text file into segments having data, or as a prototype to generate a segment of an output file.

During this operation, when a pointer to a macro is reached in the template, that pointer is used to invoke the macro class, and this macro class is used to map data from one of the segments of the text file to the computer application. The macro class then invokes another template to further process the text file. In order to help ensure that the macro invoked the appropriate template, the name of that template is passed to the macro

when that macro class is invoked to map data from the text file to the computer application. The macro class then uses that name to invoke that other template to further process the text file.

Andrews, et al.

Andrews, et al. discloses a procedure for translating source code from one high-level computer language to another. Goals of this translation procedure are to combine pieces of a source file that were generated in different translation sessions, and to ensure textual consistency of each piece of generated code in the resultant code files. In the disclosed procedure, fragment templates and partition templates are extracted from a source language text file, and a check is made for textual consistency of the target language code generated for each partition template. The described process then pieces together a target language code file from the partition templates, and combines pieces of the target language code file that were generated in different translation sessions.

One specifically disclosed translation process is referred to as the Rosetta Translator, and this process translates code from the portable Transaction Application Language (pTAL) to the C++ language. This Translator uses a syntax tree representation and a token mechanism. In particular, a source language syntax tree is used to represent the syntactic structure of a virtual source, and a source language fragment tree is employed to represent the virtual source production mechanisms that were used to create the virtual source. Tokens are placed on the leaves of the syntax tree; and each of these tokens is also used as a leaf of the fragment tree, and indicates which virtual source production mechanism brought it into the virtual source.

Fig. 6 of Andrews, et al. was specifically discussed in the Board's Decision. In Finding of Fact 9, the Board stated that Fig. 6 of Andrews, et al. shows that the macro invocation fragment invokes another partition template (labeled part'n) (Decision, Page 7, lines 21-23.).

Fig. 6 is not extensively discussed in Andrews, et al. It is noted in column 8 of Andrews, et al. that this Figure shows the second half of a fragment tree. There does not appear to be any discussion of how or when that item is included in the macro class. Even if this term "part'n" is considered to invoke another template, there is no indication that this term is passed to the macro when the macro itself is invoked.

As mentioned above, the Board stated in the Decision that "Andrews inherently discloses passing a name for another template when the macro class is invoked, and that in order "for the macro invocation fragment to invoke the partition template, a person of ordinary skill in the art would recognize that the name of the partition template must necessarily be passed to the macro invocation fragment" (Decision, page 12, lines 19-24). Even assuming for the sake of argument that this is true, it does not follow that it is inherent that the name was passed to the macro when the macro is invoked. That name, for example, could be passed to the macro before the macro is invoked.

Clearly, then, Andrews, et al. does not disclose, within the meaning of 35 U.S.C. 102, the step, expressly set forth in Claim 14, of "passing to said macro class a name for said another template when said macro class is invoked." Accordingly, Claim 14 and Claims 15 and 16 are not anticipated by Andrews, et al.

III . Conclusion

For the reasons advanced above, the Board is respectfully asked to rehear the issue of whether the final rejection of Claims 14-16 is proper. And the Board is further requested to reverse that rejection.

Respectfully submitted,

Dated: August 16, 2007

John S. Sensny
Registration No. 28,757
Attorney for Applicants

Scully, Scott, Murphy & Presser, P.C.
400 Garden City Plaza – Suite 300
Garden City, New York 11530
(516) 742-4343

JSS:jy